基于标签分类的协同过滤推荐算法 *

朱峥宇，曹晓梅

(南京邮电大学 计算机与软件学院，南京 210000)

摘　要：传统的协同过滤根据用户的行为去预测可能喜欢的产品，是当前应用最广泛的推荐算法之一。但随着用户规模的急剧扩大，有价值的信息占比较少，存在稀疏性等问题，导致推荐质量不高。针对这一问题，提出了一种基于标签分类的协同过滤推荐算法。将不完整的数据样本根据标签进行分类，使分解的矩阵依赖于类，随后使用迭代投影寻踪的方法计算类依赖矩阵的线性组合及其对应的权重。开放数据集实验表明，该方法在保持一定分类准确率的前提下，平均降低了35.23%的插补误差，优于传统协同过滤推荐算法。

关键词：协同过滤；矩阵分解；交替最小二乘法；迭代投影寻踪；监督学习

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2018.01.0065

Collaborative filtering recommendation algorithm based on label classification

Zhu Zhengyu, Cao Xiaomei

(*School of Computer & Software*, *Nanjing University of Posts & Telecommunications*, *Nanjing* 210000, *China*)

Abstract: Traditional collaborative filtering is one of the most widely used recommendation algorithms based on the user behavior. However, with the rapid expansion of the user scale, there are fewer valuable information so that it leads to bad recommendation quality because of matrix sparsity. To solve this problem, this paper proposed a collaborative filtering recommendation algorithm based on label classification. Incomplete data samples were categorized according to the labels so that the decomposed matrix could depend on the class. Then the linear combination of class-dependent matrices and its corresponding weights were calculated by using iterative projection pursuit. The experiments of open datasets show that the proposed method reduces the average interpolation error by 35.23% while maintaining certain classification accuracy. This method is better than the traditional collaborative filtering recommendation algorithm.

Key words: collaborative filtering; matrix factorization; alternating least squares; iteration projection pursuit; supervised learning

0　引言

互联网的出现与发展使人们生产、复制、传播信息的能力大大增强，人们正在面临前所未有的信息过载问题。在此背景下，推荐系统应运而生。具体而言，推荐系统是通过收集和分析用户的各种数据来学习用户的兴趣和行为模式，从而为用户在庞大的信息中推荐他所需要的信息或者服务[1]。目前互联网的几大支柱产业，包括电子商务和社交网络等，都不同程度地使用了推荐系统技术[2]。

目前，在众多的推荐算法中，协同过滤（collaborative filtering,CF）算法是应用最广泛的推荐算法之一，它根据用户-项目评分数据，计算用户（或项目）之间的相似度进行推荐。协同过滤推荐算法主要包括基于邻域和基于模型两类。基于邻域的协同过滤算法是根据用户的历史信息计算用户（产品）之间的相似性，然后根据其偏好推荐相似的用户（产品）给他。基于模型的协同过滤主要通过用户对产品的评分信息训练出相应的模型，利用模型再进行未知数据的预测。由于其良好的扩展性和可实践性，被广泛地使用并且获得巨大的成功。但是随着用户和项目数量的急剧增加，基于协同过滤的推荐系统面临用户—评分矩阵稀疏性的问题。

针对这个问题，传统方法主要是降维和基于聚类的方法，国内外研究者提出了多种解决方案。文献[3]采用主成分分析（PCA）方法对用户项目评分矩阵进行降维处理，缓解了输入数据的稀疏性问题。文献[4]提出了一种基于聚类平滑联合来减少数据稀疏的不良影响，但是这种方法丢失了部分用户评价的数据。文献[5]提出的聚类方法不能反映用户之间的兴趣差异，

收稿日期：2018-01-20；**修回日期：**2018-03-23　　**基金项目：**国家自然科学基金资助项目（61202353）；国家“973”计划资助项目（）(2011CB302903)；江苏省高校优势学科建设工程资助项目（yx002001）

作者简介：朱峥宇（1993-），男，江苏连云港人，硕士，主要研究方向为推荐系统（18651710951@163.com）；曹晓梅（1974-），女，副教授，博士，主要研究方向为网络与信息安全.

因此推荐结果的准确性并没有明显提高。文献[6]在弱关系的社交网络中，采用基于用户聚类的方法，提出两阶段聚类的推荐算法，将图摘要方法与基于内容相似度的算法结合，实现基于用户兴趣的主题推荐，有效缓解了矩阵稀疏性和冷启动的问题。

交替最小二乘法（alternating least squares， ALS）由 Zhou 等人[7]在 2008 年提出。这种方法经常用于基于矩阵分解的协同过滤推荐算法，属于基于模型的协同过滤。例如，用户评分矩阵被分解成两个矩阵，一个是用户对商品的隐含特性的偏好矩阵，另一个是隐含在商品中的特征矩阵。通过降维对缺失数据进行插补，从而进行推荐[8]。

由于在推荐系统的应用场景中，存在大量的缺失项，传统的奇异值分解等矩阵分解算法在处理数据稀疏性时存在严重的数据拟合化问题[8]，而 ALS 可以很好地解决这个问题。为了防止 ALS 模型的过度分析，相关研究在进行矩阵分解时执行了正则化。Paterek[9]通过在 cost 函数上附加额外的误差来研究 ALS 模型，同时建立矩阵因子。Zhou 等人提出了加权的 ALS 模型，其中两个岭参数在矩阵分解过程中分别施加于矩阵因子[10]。这样的做法与岭回归（ridge regression）类似。岭回归在 1962 年由 Heer 首先提出，1970 年进一步发展了该方法[11]。Ding 等人[12]开发了正交矩阵分解，也成为目前矩阵分解中比较常用的方法。

然而协同过滤推荐算法本身以及目前已知对该算法的改进，都没有在处理过程中嵌入标签信息，都属于无监督学习的范畴。当数据样本较多且稀疏的情况下，可能造成较高误插补率等问题，从而导致推荐质量不佳。

针对上述问题，本文提出了一种基于标签分类的协同过滤推荐算法（label classification based collaborative filtering recommendation algorithm, LCCF）。通过利用标签信息优势的监督学习，基于相应类的统计分布进行数据分类，用生成的类依赖替代值进行插补，以改进传统的插补方法，从而完成较为准确且质量高的推荐。

1 ALS 协同过滤推荐算法

本章先介绍初始的 ALS 协同过滤算法，然后介绍正则化 ALS 协同过滤算法。在初始 ALS 协同过滤算法研究中，矩阵分解集中在寻找基矩阵上，一个 $M*N$ 的矩阵被分解为两个低秩矩阵 U^T 和 V ，如式（1）所示，前者表示基矩阵，后者为系数矩阵。

$$X \approx U^T V \tag{1}$$

当考虑数据矩阵 X 是不缺失数据的情况，ALS 矩阵完整表示为

$$E_{ALS}(U,\ V)=\left\|X-U^T V\right\|_f^2 \tag{2}$$

这里的 U^T 和 V 分别表示 $M*D$ 和 $D*N$ 的未知矩阵；D 是中间维度；$\|\bullet\|$ 代表 Frobenius 范数；T 代表矩阵的转置。式（1）提到 U^T 和 V 都是低秩矩阵，即 $D<M$ 且 $D<N$ 。ALS 矩阵插补的目的是找到合适的 U^T 和 V ，使得损失函数 E_{ALS} 的值最小。需要注意的是，由于 U^T 和 V 是未知矩阵，如果要去计算 V ，启发式地初始化 U 是必要的，然后迭代更新 U^T 和 V 可以得到最后的融合解。此外，由于 D 是一个未知变量，需要在递归训练前进行预定义。

对于正则化 ALS 协同过滤算法，它使用岭参数来进行正则化来防止 U^T 和 V 过度拟合。岭参数的作用是稳定了逆矩阵，同时也避免了奇异矩阵的产生[13]，如式（3）所示。

$$E_{\mathrm{r}_{ALS}}(U,\ V)=\left\|X-U^T V\right\|_f^2+\rho_u\|U\|_f^2+\rho_v\|V\|_f^2 \tag{3}$$

当矩阵 X 有缺失值时，这里将缺失样本的矩阵进行点乘并得到 $G(X)$ 。原理类似计算机网络中掩码的作用，将矩阵 X 空缺数值的位置置为 0，有数值的位置保持不变。对式（3）求偏导并令等式等于 0 可得

$$V=(UU^T+\rho_u I)^{-1}U\times G(X) \tag{4}$$

同样地，

$$U=(VV^T+\rho_v I)^{-1}V\times G(X)^T \tag{5}$$

其中：I 是单位矩阵。然后迭代更新 U^T 和 V 可以得到融合解。

最后，矩阵 X 的缺失元素被生成的矩阵的相应元素所替代，即 $U^T V$ ，完成了矩阵近似，可以表示为

$$E_{\mathrm{r}_{ALS}}(U,\ V)=\left\|G(X-U^T V)\right\|_f^2 \tag{6}$$

2 LCCF 推荐算法

在第 1 章介绍的正则化 ALS 协同过滤算法基础之上，这里提出了一种基于标签分类的 LCCF 推荐算法，原理是基于矩阵分解时产生的类依赖矩阵因子的监督数据进行插补，使用类信息来创建代替值。在训练阶段，具有标签信息的不完整数据根据标签被分成不同的类别，使得矩阵依赖于类。随后，采用迭代投影寻踪的方法计算这些类依赖矩阵的线性组合及其对应的权重来对测试的数据进行插补。

2.1 对缺失数据样本进行分类

假设 X 是一个缺失数据的矩阵 $M*N$ ，y 是包含相应标签的 $N*1$ 向量，类的数量是 L 。矩阵 X 被拆分成 X_l ，其中 $l=1,..,L$ 。矩阵 X_l 的大小为 $M*N_l$ ，且 $N_1+N_2+...+N_L=N$ 。

根据式（4）和（5），在训练阶段产生的类依赖矩阵因子 U_l 和 V_l 如下：

$$V_l=(U_l U_l^T+\rho_{u_l} I)^{-1}U_l\times G(X_l) \tag{7}$$

同样地，

$$U_l=(V_l V_l^T+\rho_{v_l} I)^{-1}V_l\times G(X_l)^T \tag{8}$$

U_l 的初始化是基于类依赖的方法加上包含随机数的向量 Z ，即

$$U_l=\left[\mu_{l,1}\quad \mu_{l,2}\quad \ldots\quad \mu_{l,D}\right]^T+Z^T \tag{9}$$

其中：

$$\mu_{l,d} = \sum_{n_l=1}^{N_l} G(X_{nl}) \quad (10)$$

在式（10）中，参数 d 是第一节介绍的中间维度 D 的索引；n_l 表示类 l 样本的索引；，μ 表示矩阵的均值；$N_l *1$ 是 X_{nl} 第 n_l 列。令 t 表示不缺失值的新的 $M*1$ 样本。如果 t 属于 U_l^T 为基的线性空间时，这意味着当 U_l^T 包含所有基时，U_l^T 中向量的线性组合存在。即组合系数可用 $1*D$ 的向量 v 表示，使得 $t = U_l^T \times v_l^T$。此外，v_l^T 与 v_l 等价，原因是当训练阶段观察到足够的样本时，v_l 可以被看做在 U_l^T 所有可能的组合[14]。因此，$v_l^T = v_l \times a_l$，其中 a_l 是一个 $N_l *1$ 的系数向量。

因为 U_l^T 和 V_l 都是近似低秩的，所以目标是找到 a_l，使得 $U_l^T V_l a_l$ 与 t 足够近似，即

$$t \approx U_l^T V_l a_l = U_l^T v_l^T \quad (11)$$

系统可以通过测量重构误差来确定定新样本的类别，即新加入的样本数据也可以进行基于类的插补，从而不需要冗余的再进行数据训练。令重构差为 e_l，即

$$e_l = t - U_l^T v_l^T \quad (12)$$

其中：

$$v_l^T = (U_l U_l^T + \rho_l I)^{-1} U_l t \quad (13)$$

随着递归地进行重构，系统可以对测试样本进行近似计算并进行分类，即式（13）为最后改进的岭回归（RR）的解，最后可以得出式（14）的目标解。

$$E_{RR}(v_l) = \left\| t - U_l^T v_l^T \right\|_f^2 + \rho_l \left\| v_l^T \right\|_f^2 \quad (14)$$

2.2 使用迭代投影寻踪进行数据插补

当需要处理的数据维数较高时，数据结构常表现在几个重要的投影方向上。投影寻踪方法可以有效地发现高维数值的结构和特征[15, 16]。

因此在算法中提出了基于岭回归的迭代投影寻踪方法。迭代投影寻踪方法可以迭代地检测由类依赖基矩阵形成的向量之间的最近距离，并且也可以检测不完整的向量。对于具有缺失数据的 t，需要在上述过程中执行数据插补和分类。程序思想如下：

a) 初始化 $\hat{t}_l$，用 0 来填充缺失数据的 t。

b) 基于每个类计算 v_l。其中 i 代表迭代次数。

$$v_l^T[i] = (U_l U_l^T + \rho_l I)^{-1} U_l \hat{t}_l[i] \quad (15)$$

c) 通过重构 t 来估算缺失值。其中运算符 $\oplus$ 表示用估计值 $\hat{t}$ 代替实际缺失值。

$$\hat{t}_l[i] = U_l^T \times v_l^T[i] \quad (16)$$

$$\hat{t}_l[i+1] = t \oplus \hat{t}_l[i] \quad (17)$$

d) 重复步骤 b)～d)，直到重构误差 e 收敛。

$$e_l[i+1] = G\left(t - \hat{t}_l[i+1]\right) \quad (18)$$

在第 3 章实验中，使用 RMSE 作为停止训练的标准。RMSE 表示均方根误差，其中 $e_{l,m}$ 表示类 l。

$$RMSE = \sqrt{\frac{\sum_{m=1}^{M} e_{l,m}^2[i+1]}{M}} \quad (19)$$

e) 通过选择具有最小 RMSE 的 l 可以确定预测类。

$$l^* = \arg_l \min(RMSE) \quad (20)$$

3 实验结果及分析

3.1 数据集及预处理

为了检验基于标签分类的协同过滤推荐算法与传统的正则化 ALS 协同过滤算法之间推荐质量的差别，并且为了避免单一数据集出现过拟合化的问题，本次实验使用的是明尼苏达大学 GroupLens 小组提供 MoiveLens 数据集和在线视频提供商 Netflix 提供的 Netflix 数据集，这些数据集是用户对电影的真实评分数据，并且每个电影都有相应的类别字段，包含电影 ID、电影名称、电影类型等，这些数据集的信息如表 1 所示。

表 1 实验数据集

名称	用户	电影	评分数量
ML-100K	6 040	3 592	100 000
Netflix	13 682	7 862	650 000

实验根据数据集标签对电影类别进行建模，并且采用随机抽取 80%的数据进行训练，其余的 20%用于测试，这样可以保证训练数据与测试数据都是随机的且都来自同一数据集。对于岭参数 ρ_u、ρ_v、ρ_l 通常都设为 0.5。此外，在训练阶段，基于标签和传统的协同过滤的平均 RMSE 阈值都设置为 0.01。

这里将数据集送入支持向量机（SVM）分别进行有监督和无监督的进行训练。SVM 是由 Vanpik 领导的 AT&TBell 实验室小组在 1963 提出的一种新的并且非常有潜力的分类技术，目前主要用于模式识别领域[17]。SVM 的关键在于核函数，因为它通过将数据映射到高维空间，来解决在原始空间中线性不可分的问题，避免了直接在高维空间中的复杂计算[18]。核函数的类型主要有线性函数、多项式函数和径向基函数（RBF）。目前实际应用最广泛的是 RBF 核。与多项式核函数相比，当多项式的阶数较高时，会出现核矩阵元素趋于无穷大或者无穷小的问题，而 RBF 会减少数值的计算困难，线性函数是 RBF 的特例，大部分情况下 RBF 的适用范围更广。综合考虑，本文实验采用 RBF 核函数中比较常用的插值法，通过选择合适的插值半径进行实验。由于 Netflix 相对比 ML-100K 数据集样本数量比较多，RBF 插值半径分别设置为 1.00 和 10.00。

3.2 评价标准

本文实验使用均方根误差 RMSE 作为评价标准。RMSE 通过计算预测的用户评分与实际的用户评分之间的偏差来度量预测的准确性，是最常用的一种推荐质量度量方法。一般 RMSE 越小，意味着推荐的质量越高。结合第 2 章提出的迭代投影寻踪方法，RMSE 表示为

$$RMSE=\sqrt{\frac{\sum_{m=1}^{M} e_{l,m}^{2}\left[i+1\right]}{M}} \tag{21}$$

本文还使用分类准确率（classification accuracy）作为评价标准。准确率是评价一个分类算法好坏比较直观的标准，只有保证一定的分类准确率前提下，2.2 节 LCCF 算法下的矩阵插补才有意义。分类准确率表示为

$$Accuracy=(TP+TN)/(P+N) \tag{22}$$

3.3 实验结果及分析

本文实验将分为两个部分：第一个部分是将传统的 ALS 协同过滤算法（简称 ALS-CF）与提出的 LCCF 算法进行分类准确率的对比实验；第二部分对比它们的均方根误差，即 RMSE，最后得出实验结论。

3.3.1 计算分类准确率

对于分类准确率的对比实验，本文将 ALS-CF 和 LCCF 分别送入 SVM 进行训练，计算出相应的分类准确率。实验结果如图 1 和 2 所示。

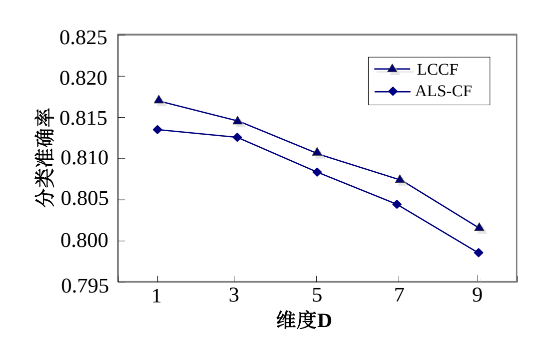


图 1　ML-100k 数据集的分类准确率

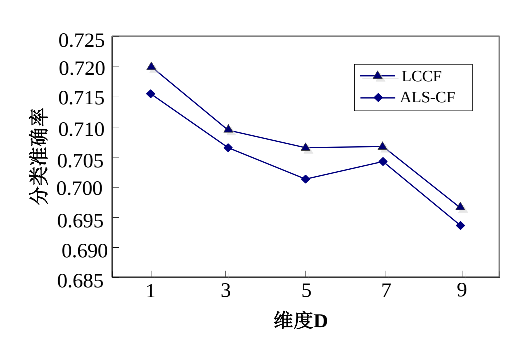


图 2　Netflix 数据集的分类准确率

实验结果表明，对于 ML-100K 数据集，LCCF 算法平均分类准确率为 81.21%，ALS-CF 算法平均分类准确率为 80.63%；对于 Netflix 数据集，LCCF 算法平均分类准确率为 71.54%，ALS-CF 算法平均分类准确率为 70.62%。

结合图 1 和 2，无论是 ML-100K 数据集还是 Netflix 数据集，即使在不同维度 D 下，LCCF 算法的分类准确率都略高于传统的 ALS-CF，保持了一定的分类准确率。也可以看出，维度的不同对于分类准确率也是有一定的影响，大部分都呈现维度越大，准确率越低的趋势。对比图 1 和 2 可以看出，图 1 分类准确率是高于图 2 的分类准确率，这与数据集的稀疏性程度有关，Netflix 数据集较 ML-100K 数据集稀疏许多。另外图 2 中 Netflix 数据集维度为 7 时，准确率略高于维度 5，这里在 SVM 中使用多项式核函数进行二次验证，实验结果与使用 RBF 核函数类似，都出现维度 7 的分类准确率高于维度 5 的情况，这样的情况是允许的。

3.3.2 计算均方根误差 RMSE

第二部分实验对比 ALS-CF 与 LCCF 算法的平均 RMSE。首先将两个数据集的空缺率分别设置为 10%、20%和 30%，然后送入 SVM 进行训练，最后计算出均方根误差 RMSE。这里设置空缺率的目的是为了验证不同稀疏性的情况下，所提出的 LCCF 算法是否依然可以降低 RMSE，提高推荐质量。实验结果如图 3 和 4 所示。

实验结果表明，无论是 ML-100k 数据集还是 Netflix 数据集，所提出的 LCCF 算法的 RMSE 较 ALS-CF 算法更小。以 ML-100K 数据集为例，ALS-CF 算法的平均 RMSE 为 0.070 492，而 LCCF 算法的平均 RMSE 为 0.035 261，该方法平均减少插补误差 35.23%，从而提高了推荐质量，优于传统的 ALS-CF 算法。

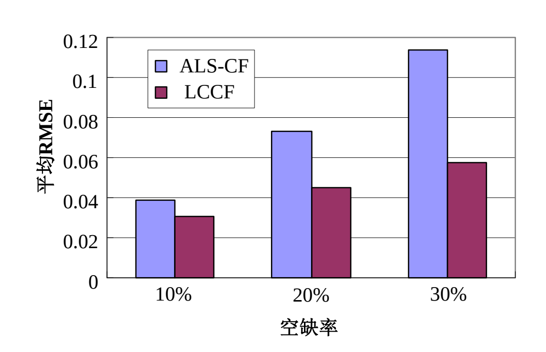


图 3　ML-100k 数据集的平均 RMSE

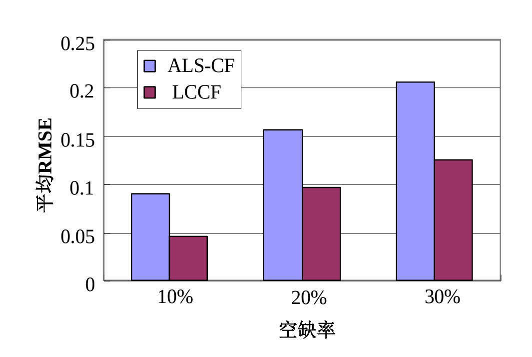


图 4　Netflix 数据集的平均 RMSE

4　结束语

本文提出了基于标签分类改进的协同过滤 LCCF 算法，空缺数据根据类依赖基矩阵进行插补，可以从系数矩阵导出依赖于类的回归权重，对新的不完全样本进行分类和插补。为了找到合适的权重，本文使用了迭代投影寻踪的方法。该方法递归

地检查由类依赖矩阵形成的向量与不完整向量之间的最近距离。接着与正则化 ALS 协同过滤推荐算法进行对比。实验结果表明，所提出的改进方法比典型的协同过滤在保持一定的分类精准度的前提下能有效地减少插补误差，能够进行更有效更精确的推荐。然而这种基于标签分类的方法只适用于已存在标签信息的数据集。推荐系统中的数据普遍存在标签数少和用户数庞大等问题，所以下一步将研究一种适应性更好的混合推荐算法，解决其他情况下的数据稀疏性等问题。

参考文献：

[1] Zhao Z D, Shang M S. User-based collaborative filtering recommendation algorithms on hadoop [C]// Proc of the 3rd International Conference on. Knowledge Discovery and Data Mining, 2010. 2010: 478-481.

[2] Mehmood A, Natgunanathan I, Xiang Y, *et al.* Protection of big data privacy [J]. IEEE Access, 2016, 4: 1821-1834.

[3] Mnih A, Salakhutdinov R R. Probabilistic matrix factorization [C]// Advances in Neural Information Processing Systems. 2008: 1257-1264.

[4] Yin C X, Peng Q K. A careful assessment of recommendation algorithms related to dimension reduction techniques [J]. Knowledge-Based Systems, 2012, 27: 407-423.

[5] Shepitsen A, Gemmell J, Mobasher B, *et al.* Personalized recommendation in social taggig systems using hierarchical clustering [C]// Proc of ACM Conference on Recommender Systems. 2008: 259-266.

[6] 陈克寒，韩盼盼，吴健. 基于用户聚类的异构社交网络推荐算法 [J]. 计算机学报, 2013, 36 (2): 349-359. (Chen Hanke, Han Panpan, Wu Jian. Heterogeneous social network recommendation algorithm based on user clustering [J]. Chinese Journal of Computers, 2013, 36 (2): 349-359.)

[7] Zhou Y, Wilkinson D, Schreiber R, *et al.* Large-scale parallel collaborative filtering for the netflix prize [J]. Lecture Notes in Computer Science, 2008, 5034: 337-348.

[8] Anagnostopoulos C, Triantafillou P. Scaling out big data missing value imputations: Pythiavs. godzilla [C]// Proc of ACM SIGKDD InternationalConference on Knowledge Discovery and Data Mining. 2014: 651-660.

[9] Paterek A. Improving regularized singular value decomposition for collaborative filtering [C]// Proc of KDD Cup and Workshop. 2007: 5-8.

[10] Pennington J, Socher R, Manning C. Glove: Global vectors for word representation [C]// Proc of Conference on Empirical Methods in Natural Language Processing. 2014: 1532-1543.

[11] Gruber M. Improving efficiency by shrinkage: the james-stein and ridge regression estimators [M]. 2017.

[12] Ding C, Li T, Peng W, *et al.* Orthogonal nonnegative matrix t-factorizations for clustering [C]// Proc of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2006: 126-135.

[13] Kung S Y. Kernel methods and machine learning [M]. [S. l.] : Cambridge University Press, 2014.

[14] Clancey K F, Gohberg I. Factorization of matrix functions and singular integral operators [M]. 2013.

[15] Friedman J H, Stuetzle W. Projection pursuit regression [J]. Journal of the American Statistical Association, 1981, 76 (376): 817-823.

[16] Ferraty F, Goia A, Salinelli E, *et al.* Functional projection pursuit regression [J]. Test, 2013, 22 (2): 293-320.

[17] Joachims T. Making large-scale SVM learning practica, Technical Report, SFB 475 [R]. [S. l.] : Universität Dortmund, 1998.

[18] Gu B, Sheng V S, Tay K Y, *et al.* Incremental support vector learning for ordinal regression [J]. IEEE Trans on Neural Networks and Learning Systems, 2015, 26 (7): 1403-1416.